



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.				
10/667,757	09/22/2003	Udayan Rajendra Kanade	COT-003	1371				
<div>7590 William L. Botjer PO Box 478 Center Moriches, NY 11934</div>								
<div>05/29/2007</div>								
<div>EXAMINER WILSER, MICHAEL P</div>								
<table border="1"><thead><tr><th>ART UNIT</th><th>PAPER NUMBER</th></tr></thead><tbody><tr><td>2109</td><td></td></tr></tbody></table>					ART UNIT	PAPER NUMBER	2109	
ART UNIT	PAPER NUMBER							
2109								
<table border="1"><thead><tr><th>MAIL DATE</th><th>DELIVERY MODE</th></tr></thead><tbody><tr><td>05/29/2007</td><td>PAPER</td></tr></tbody></table>					MAIL DATE	DELIVERY MODE	05/29/2007	PAPER
MAIL DATE	DELIVERY MODE							
05/29/2007	PAPER							

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/667,757

Applicant(s)

KANADE, UDAYAN RAJENDRA

Examiner

Michael Wilser

Art Unit

2109

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 22 September 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 22 September 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>9/22/03</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the original filing of September 22, 2003. Claims 1-18 are pending and have been considered below.

Specification

2. The disclosure is objected to because of the following informalities: the examiner notes the use of acronyms (e.g. DSP, DMA, etc.) throughout the specification without first providing a description in plain text, as required.
3. On page 8, lines 16-17 of the specification the applicant refers to the application program 112. However, on page 7, line 14 the applicant refers to the reference number 112 as referring to the local program store managing service. Additionally, on page 7, line 15 the applicant refers to the above mentioned application program as reference character 116. For purpose of examination the examiner is interpreting the applicant to have meant to use reference character 116 on line 17 of page 8.
4. On page 8, line 17 of the specification the applicant refers to the interacting threads 114. However, on page 7, line 15 the applicant refers to the reference number 114 as referring to the other services. Additionally, on page 7, line 16 the applicant refers to the above mentioned interacting threads as reference character 118. For purpose of examination the examiner is interpreting the applicant to have meant to use reference character 118 on line 17 of page 8.

Art Unit: 2109

5. On page 9, lines 1-2 of the specification the applicant refers to the special-purpose processor 202. However, on page 7, line 25 the applicant refers to the reference number 202 as referring to the storage area. Additionally, on page 7, lines 9-10 the applicant refers to the above mentioned special-purpose processor 104. For purpose of examination the examiner is interpreting the applicant to have meant to use reference character 104 on line 2 of page 9.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. Examiner's Note. The Applicant appears to be attempting to invoke 35 U.S.C. 112 6th paragraph in Claims 17 and 18 by using "means-plus-function" language. However, the Examiner notes that the only "means" for performing these cited functions in the specification appears to be computer program modules. While the claims pass the first test of the three-prong test used to determine invocation of paragraph 6, since no other specific structural limitations are disclosed in the specification, the claims do not meet the other tests of the three-prong test. Therefore, 35 U.S.C. 112 6th paragraph has not been invoked when considering these claims below.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

Art Unit: 2109

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over McDonald (US 7,159,216) in view of Spoltore et al. (US 2004/0015971).

Claim 1: McDonald discloses a method for allocating processor to threads comprising:

- a. compiling an application program in response to a request for execution of the application program (column 3, lines 3-46);
- b. the application program comprising interacting threads (column 3, lines 3-46);
- c. each thread capable of independently executing an application segment (column 3, lines 3-46);
- d. scheduling threads on various processors based on type and availability of the processors (column 11, lines 62-67 & column 12, lines 1-3); and
- e. managing program stores of each processor for complying with processor load (column 13, lines 13-55).

However, McDonald does not explicitly disclose of the processing load being dependent on the requests for specific programs and the frequency of the requests. However, Spoltore discloses a similar method for allocating processors in which the load does depend on the frequency and the type of request (page 2, paragraph 29). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have balance the load in McDonald according to frequency and type of request. One would have been motivated to balance the load by frequency and type of

Art Unit: 2109

request to minimize the amount of idle time in the system and improve overall processing speed.

Claim 2: McDonald and Spoltore disclose a method as in Claim 1 above, and McDonald further discloses a request queue which stores all of the stored threads that are waiting to be allocated to a processor (column 10, lines 4-43).

Art Unit: 2109

Claim 3: McDonald and Spoltore disclose a method as in Claim 3 above, and McDonald further discloses:

- a. allocating a free processor to a thread that does not request access to a program stored in the processor (column 10, lines 4-43);
- b. allocating a free processor to a thread requesting access to program stored in the processor (column 1, lines 13-55); and
- c. stalling the requesting thread and adding it to the tail of the request queue if no processor is available (column 10, lines 4-43).

Claim 4: McDonald and Spoltore discloses a method as in Claim 3 above, and McDonald further discloses that a thread requesting a program already loaded in a processor will be switched from the current processor to the processor holding the program (column 11, lines 62-67 & column 12, lines 1-3).

Claim 5: McDonald and Spoltore disclose a method as in Claim 3 above, and McDonald further discloses:

- a. receiving an allocation request from a thread for a processor with a program loaded on it (column 10, lines 4-43);
- b. searching for a free processor with the requested program loaded on it (column 11, lines 47-61);
- c. allocating the free processor to the with program loaded to the requesting thread (column 11, lines 62-67 & column 12, lines 1-3); and

d. loading the requested program onto a free processor and allocating it to the requesting thread if no other processors are available (column 12, lines 25-58).

Claim 6: McDonald and Spoltore discloses a method as in Claim 1 above, and McDonald further discloses of allocating free processor to a thread that requests program already loaded on the processor (column 13, lines 13-55). Furthermore, Spoltore further discloses of removing programs from the free processor until enough space is available to load the program requested by the thread (pages 3 & 4, paragraphs 42 & 43). Therefore it would have been obvious to one having ordinary skill in the art at the time of invention to remove programs from the processor until space was available in McDonald. One would have been motivated to remove programs from the free processor so that the thread that the processor was allocated to could load and run the program that it needs to continue its processing needs.

Claim 7: McDonald discloses a method for allocating processor to threads comprising:

Art Unit: 2109

- a. receiving an allocation request from a thread for a processor with a program loaded on it (column 10, lines 4-43);
- b. allocating the free processor to the with program loaded to the requesting thread (column 11, lines 62-67 & column 12, lines 1-3);
- c. stalling the requesting thread and adding it to the tail of the request queue if no processor is available (column 10, lines 4-43);
- d. checking the request queue for pending requests once a processor has been released by the requesting thread (column 13, lines 13-55);
- e. allocating the free processor to the first thread in the request queue that requests the program loaded on it (column 13, lines 13-55); and
- f. allocating free processor to first thread in request queue if no thread in queue requests program loaded on processor (column 11, lines 62-67 & column 12, lines 1-3).

However, McDonald does not explicitly disclose of receiving control of the allocated processor from the requesting thread once the processor becomes idle. However, Spoltore discloses a similar method for allocating processors in which control of the processor is received once the processor becomes idle (page 3, paragraph 35). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have the processor in McDonald return control upon becoming idle. One would have been motivated to return control of the processor so that the CPU could then assign the idle processor to another thread waiting to be executed.

Art Unit: 2109

Claim 8: McDonald and Spoltore disclose a method as in Claim 7 above, and McDonald further discloses:

- a. searching for a free processor with the requested program loaded on it (column 11, lines 47-61);
- b. of allocating free processor to a thread that requests program already loaded on the processor (column 13, lines 13-55); and
- c. . loading the requested program onto a free processor and allocating it to the requesting thread if no other processors are available (column 12, lines 25-58).

Claim 9: McDonald and Spoltore disclose a method as in Claim 8 above, and McDonald further discloses:

- a. loading the requested program into space created on the processor (column 11, lines 47-61); and
- b. allocating the processor to the requesting thread (column 10, lines 40-59).

Furthermore, Spoltore further discloses of virtually evicting and actually evicting programs from a processor to create room for program of the requesting thread (pages 3 & 4, paragraphs 42 & 43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have both virtually and actually evicted programs from processors in McDonald. One would have been motivated to remove programs from the free processor so that the thread that the processor was allocated to could load and run the program that it needs to continue its processing needs.

Art Unit: 2109

Claim 10: McDonald and Spoltore disclose of a method as in Claim 9 above, and McDonald further discloses that evicting programs is done by LRU, LFU, or FIFO (column 15, lines 40-59).

Claim 11: McDonald and Spoltore disclose a method as in Claim 9 above, and Spoltore further discloses that evicting programs comprise the use of task information regarding priority, execution time, pending time, and relevance (pages 3 & 4, paragraphs 42 & 43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have used information pertaining to the tasks to evict programs in McDonald. One would have been motivated to use task information to evict tasks so that higher priority tasks are executed before lower priority tasks and to ensure that lower priority tasks eventually are performed and are not starved.

Claim 12: McDonald and Spoltore disclose a method as in Claim 7 above, and McDonald further discloses:

- a. loading the requested program into space created on the processor (column 11, lines 47-61); and
- b. allocating the processor to the requesting thread (column 10, lines 40-59).

Furthermore, Spoltore further discloses of virtually evicting and actually evicting programs from a processor to create room for program of the requesting thread (pages 3 & 4, paragraphs 42 & 43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have both virtually and actually evicted

Art Unit: 2109

programs from processors in McDonald. One would have been motivated to remove programs from the free processor so that the thread that the processor was allocated to could load and run the program that it needs to continue its processing needs.

Claim 13: McDonald and Spoltore disclose of a method as in Claim 12 above, and McDonald further discloses that evicting programs is done by LRU, LFU, or FIFO (column 15, lines 40-59).

Claim 14: McDonald and Spoltore disclose a method as in Claim 12 above, and Spoltore further discloses that evicting programs comprise the use of task information regarding priority, execution time, pending time, and relevance (pages 3 & 4, paragraphs 42 & 43). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have used information pertaining to the tasks to evict programs in McDonald. One would have been motivated to use task information to evict tasks so that higher priority tasks are executed before lower priority tasks and to ensure that lower priority tasks eventually are performed and are not starved.

Claim 15: McDonald and Spoltore disclose a method as in Claim 7 above, and Spoltore further discloses that receiving control of a processor from a thread is embodied in a computer program product (page 1, paragraph 10).

Claim 16: McDonald discloses a system for allocating processor to threads comprising:

a. compilation service for compiling application program in response to a request for execution of the application program (column 3, lines 3-46);

b. application program comprising a plurality of interacting threads (column 3, lines 3-46); and

c. processor allocation service for scheduling and synchronizing threads to various processors (column 5, lines 50-67).

However, McDonald does not explicitly disclose a program store for managing local programs for each of the processors complying with the processor load. However, Spoltore discloses a similar system for allocating processors which does comprise a program stores for managing local programs of the processors (page 3, paragraphs 36 & 37). Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention to have a program store in McDonald for managing local programs. One would have been motivated to have a local store to manage local programs so that programs loaded onto processor can be easily identified and therefore improve computing time and throughput of the system.

Claim 17: McDonald and Spoltore disclose a system as in Claim 16 above, and McDonald further discloses:

a. allocating a free processor to a thread that does not request access to a program stored in the processor (column 10, lines 4-43);

b. allocating a free processor to a thread requesting access to program stored in the processor (column 1, lines 13-55); and

c. stalling the requesting thread and adding it to the tail of the request queue if no processor is available (column 10, lines 4-43).

Claim 18: McDonald and Spoltore discloses a system as in Claim 16 above, and McDonald further discloses of allocating free processor to a thread that requests program already loaded on the processor (column 13, lines 13-55). Furthermore, Spoltore further discloses of removing programs from the free processor until enough space is available to load the program requested by the thread (pages 3 & 4, paragraphs 42 & 43). Therefore it would have been obvious to one having ordinary skill in the art at the time of invention to remove programs from the processor until space was available in McDonald. One would have been motivated to remove programs from the free processor so that the thread that the processor was allocated to could load and run the program that it needs to continue its processing needs.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.


- a. Yokoya (US 6,199,093) Processor Allocating Method/Apparatus in Multiprocessor System, and Medium for Storing Processor Allocating Program.
- b. Benhase et al. (US 7,178,147) Method, System, and Program for Allocating Processor Resources to a First and Second Types of Tasks.

c. Chung et al. (US 2004/0194098) Application-Based Control of Hardware Resource Allocation.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Wilser whose telephone number is (571) 270-1689. The examiner can normally be reached on Mon-Fri 7:30-5:00 EST (Alt Fridays Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.


MPW
May 24, 2007


James Myhre
Supervisory Patent Examiner